Edward Park

Sep.1, 2020

Intro to Programming 110A

Assignment #8

Assignment 8: Classes

**Introduction**

In the eighth assignment, there were two parts:

1. We looked at classes and the various ways to modify and use them through properties constructors, destructors, attributes, and getter and setter properties.
2. More in depth use and navigation of GitHub via the desktop version.

**Assignment Part 1:**

At this point, I thought I was irretrievably lost, but in soldiering through this exercise, managed to piece together enough to do at least some of the assignment. Ignoring the sad, broken, bleeding animal that is my Assigment 8 code, I’d like to focus on sunnier vistas and go straight to what I’ve learned.

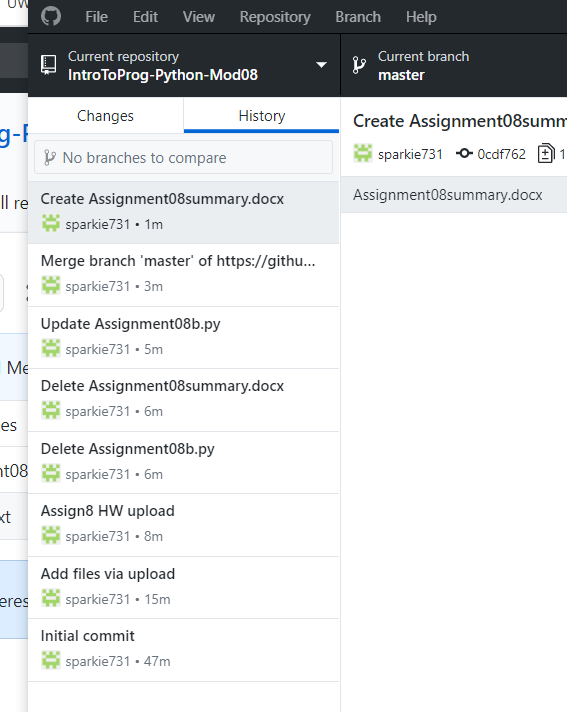
A big plus is that whereas before it felt like I was panicking and being physically drowned in a sea of code, that status has improved greatly to something like being gently suffocated by a child and knowing I’ll survive- a surmountable challenge with a little rage for motivation.

In an attempt to think out loud to clarify my issues, here’s what I understand:

1. Classes are the overarching category that can consist of any number of functions and/or objects that are separated out to perform a task. This is a similar, but separate distinction from what we had separation of duties as “Data”, “Processing” and “Presentation”.
2. Classes are further separated into:
   1. Fields
      1. Creating/naming the variables (i.e. “fields” to be used)
   2. Constructors (with a subfield of Attributes)
      1. The discussion of “passing” something into the constructors with the \_\_init\_\_ is not clear yet to me, as is the bit about something being invisible.
      2. Sounds like you create an object in the class to treat a class like an object, and add the step of also creating new implicit variable names for the explicit Fields that were initially created.
   3. Properties: Consists of Getters and Setters, and determined the properties of the Fields and Attributes above.
      1. Getters: controls the formatting of the data like capitalizations, etc.
      2. Setters: validates the data (i.e. exception handling, verifying string/float/integer)
   4. Methods: Functions
3. Functions within classes are accessed by the format “*Class.Function*”
4. Questions: What is the meaning of the double underscore being used for “private” mean? So far, my understanding of using something “indirectly” is to feed an external variable into it, but that doesn’t seem to make sense in this context.
5. I understood the docstring item, but did not fully grasp the significance of the @staticmethod.
6. I did manage to get most of the basic functions working (I think), but left a few unsolved items, some of which are totally critical. I am struggling with saving and writing the file, and retrieving them in a way that makes sense. It truly feels like I’m inches away from grasping it, but something about the process is confusing.

**Assignment Part 2: GIthub Desktop**

1. This was luckily far more straightforward, and actually resolved some of my confusion from before. The syncing of the desktop and online GitHub information makes it much easier to manage the two (since they will align). This reminded me that most of the software I’ve learned in my time at work is not from actually reading a manual or listening to training, but from fiddling with it and knowing IT has my back.

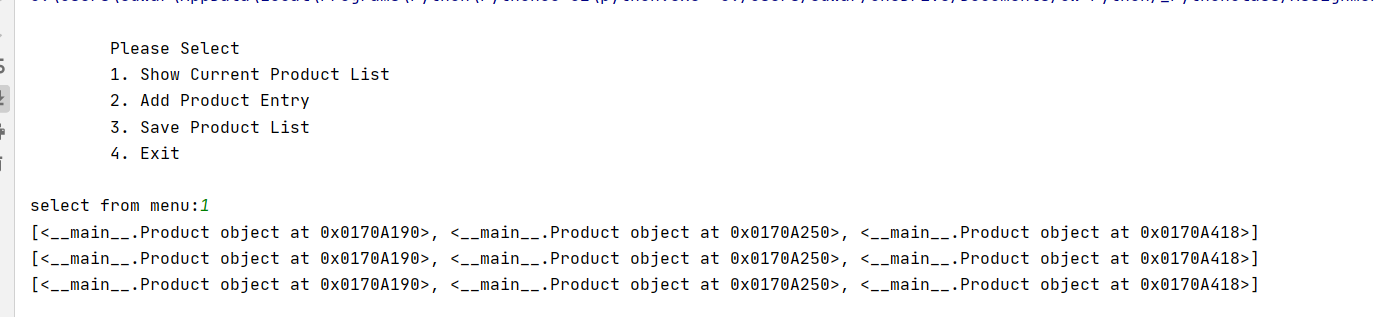


As you can see, I tried a few things like pulling in from the web site, then pushing it back from the desktop.

**Summary**

My incomprehension of some key nuances notwithstanding, I generally get what the structure of the code is meant to do, and really like how it keeps the main script uncluttered so I can use it almost as an index. It really makes the tracing and step-by-step programming very clean and organized.

Again, I think I failed to make a complete working code, and basically timed out while trying to debug it. Interestingly, some of the bugs are truly weird and I don’t know how I didn’t get them before. For example, in one case (which I never solved) the function reads from the file but then only show me the memory location, which I wouldn’t know how to do if I tried.



Another but that ended up being a huge time sink was when I realized Pycharm was not reading the text file from the same folder, and there was no way to really “re-sync” it.

On the plus side, I only seem to be a session and a half behind- I’m sort of getting the gist of what to do, and seeing other people’s code (without copying it, but just understanding it) is helping, especially for troubleshooting things like where to stick commas and colons and things.

Another positive from that exercise is that now I’m able to look at someone’s code and figure out what they’re trying to do, which may be itself a very practical skill and aid me in my path to regaining some self respect and dignity after struggling so mightily with “introductory” programming.